

## REMARKS

Claims 1-32 remain pending in the application. Reconsideration is respectfully requested in view of the following remarks.

### Section 103(a) Rejections:

The Examiner rejected claims 1-9, 13-24 and 28-33 under 35 U.S.C. § 103(a) as being unpatentable over AAPA in view of Jin et al. (U.S. Patent 6,330,689) (hereinafter “Jin”), and claims 10-12 and 25-27 as being unpatentable over AAPA in view of Jin and further in view of Crites et al. (U.S. Patent 6,097,380) (hereinafter “Crites”). Applicants respectfully traverse these rejections for at least the following reasons.

The rejection is improper because the art relied upon by the Examiner explicitly teaches away from a distributed data system comprising a plurality of nodes including at least one in-process node and at least one out-of-process node, as recited in claim 1. In rejecting claim 1, the Examiner acknowledges that AAPA (the background section of the present application) “does not explicitly teach wherein the plurality of nodes in the system comprises at least one in-process node and at least one [out-of-]process node”. In fact, as pointed out in Applicants’ response to the previous Office Action, the background section clearly and unequivocally teaches: “Conventional systems allow *only one type of configuration* - either every node is an in-process node or every node is an out-of-process node. For example, if an out-of-process client is desired, then all other clients would also need to be configured as out-of-process clients” (Page 4, lines 20 – 23). See also, with respect to Figure 1: “This configuration is referred to as out-of-process and all nodes in the distributed data system in this configuration are configured out-of-process”(Page 1, line 31 – page 2, line 3) and with respect to Figure 2: “This configuration is called in-process and all nodes in the distributed data system in this configuration are configured in-process” (Page 3, lines 16 – 18). Thus, the AAPA reference relied upon by the Examiner in rejecting claim 1 explicitly teaches away from the combination of limitations recited in claim 1. “References that teach away cannot

serve to create a *prima facie* case of obviousness.” *In re Gurley*, 27 F.3d 551, 553, 31 USPQ2d 1131, 1132 (Fed. Cir. 1994). Applicants note that on page 14 of the Final Action, the Examiner acknowledges that “a prior art reference must be considered in its entirety, including portions that would lead away from the claimed invention”. Applicants respectfully submits that when the AAPA reference is taken in its entirety, it clearly and unequivocally teaches away from the combination of limitations recited in claim 1, and therefore cannot create a *prima facie* case of obviousness.

Further with respect to claim 1, the Examiner asserts that “Jin teaches a server architecture wherein application can be run either in-process or out-of-process with the server program (see abstract)”, and that “Jin teaches a data system comprising wherein the plurality of nodes in the system comprises at least one in-process node and at least one process node (col. 6 lines 42 – 48).” The Examiner then further asserts that “it would have been obvious to one of ordinary skill in the art to modify AAPA to include both in-process and out-of-process nodes as in Jin. One would be motivated to do so to offer the flexibility to run either or both in-process and out-of-process applications (Jin, col. 6 lines 56 – 57).” Applicants strongly traverse the Examiner’s assertions for a number of reasons.

First, as pointed out in Applicant’s previous response, contrary to the Examiner’s suggestion, Jin does not teach or suggest an in-process node of a distributed data system, where the in-process node comprises a client and a distributed data manager configured to execute within the same process, as recited in claim 1. In contrast, Jin teaches “application managers” that may run “within a server’s process (i.e., in-process)” or “within separate processes (i.e., out-of-process)”. The Examiner does not rebut Applicant’s argument with respect to the lack of support in Jin for the rejection – see, e.g., page 15 of the Final Action. Instead, the Examiner once again refers to alleged teachings of the AAPA reference as reciting the claimed limitations. However, as pointed out above, since the AAPA reference teaches away from the claimed combination of limitations in claim 1, the AAPA reference cannot be relied upon to create a *prima facie* case of obviousness. The Examiner asserts on page 15 that “one

cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references”. However, Applicants only discuss the references individually to show that the Examiner’s reliance on the individual references is misplaced. It is clear that the references, whether considered alone or in combination, do not teach or suggest a distributed data system comprising a plurality of nodes including at least one in-process node and at least one out-of-process node, as recited in claim 1. Since the AAPA reference explicitly teaches away from the claimed combination of limitations, and since the Examiner appears to agree that Jin does not teach the claimed limitations under discussion, the art cited by the Examiner, when considered in its totality, fails to render the claimed combination of limitations obvious.

Further with respect to claim 1, Applicant pointed out in the previous response that there is no teaching or suggestion in the art of record that Jin’s in-process and out-of-process “application manager” and “server” combinations would in any way apply to the configuration of clients and distributed data managers in the nodes of AAPA. The Examiner does not appear to address this argument in the Final Action. Applicants submit that Jin’s use of the terms “in-process” and “out-of-process” refers to the manner in which Jin’s “application managers” responsible for processing client requests interact with Jin’s “server processes”. **Jin does not teach or suggest a client executing within the same process as a distributed data manager, much less an in-process node of a distributed data system at which a distributed data manager and a client execute within the same process, as recited in claim 1.**

**Furthermore, Jin is silent with respect to the use of “non-serialized format” and “serialized format” for communication with different entities by a distributed data manager of an in-process node, as also recited in claim 1.**

Still further with respect to claim 1, as also pointed out in Applicants’ previous response, even if the teachings of Jin were to be combined with those of AAPA, this would only lead to “application managers” (client request processors) that may “run within the same process” or “different processes” as a server, not to a distributed data

system in which at some nodes, clients run in the same process as distributed data managers, and at other nodes, clients run in separate processes from distributed data managers, as recited in claim 1. Applicants note that on page 16 of the Final Action, the Examiner suggests that one would be motivated to combine the teachings of AAPA and Jin because this would “offer developers the flexibility to run either or both in-process and out-of-process applications (Jin, col. 6 lines 55-57), thereby providing higher performance at the risk of crashing the system and affording higher reliability (Jin, col. 4, lines 55-62). The Examiner is incorrect. AAPA expressly teaches away from the combination proposed by the Examiner. As noted above, APA teaches that in conventional distributed data systems, all nodes must be either in-process or out-of-process (no mixing of in-process and out-of-process nodes is allowed). It is improper to combine references where the references teach away from their combination. *In re Grasselli*, 218 USPQ 769, 779 (Fed. Cir. 1983). Therefore, the rejection is clearly improper.

With respect to independent claims 17 and 32, in each case the Examiner once again acknowledges in the Final Action that AAPA “does not explicitly teach in-process node and out-of-process node are in the distributed data system” and asserts that the limitations of the claims are rendered obvious by the combination of AAPA with the same portions of Jin that were cited in the rejection of claim 1. The Examiner’s assertions with respect to claim 1 have already been shown to be in error. Accordingly, the rejection of claims 17 and 32 is also unsupported by the cited art.

With respect to claim 6, the Examiner alleges that the limitation “wherein all data store operations performed by the distributed data manager in the in-process node store data in a non-serialized object format in a data store of the in-process node” is taught in AAPA page 4, lines 15 – 18 and lines 8 – 10. **The Examiner is incorrect.** Lines 15 – 18 of page 4 teach that “in-process configuration data may be communicated between a distributed data manager and a client sharing the same process space, without the additional computation requirement for serialization/deserialization” and lines 8 – 10 teach that “a requesting distributed data manager stores data in its data store and returns

the data or a pointer to the client 201A indicating where the requested data is in the data store 221A”, but nowhere in the cited art is there a teaching or suggestion of all data store operations performed by a distributed data manager in an in-process node storing data in a non-serialized format in a data store of the in-process node, as recited in the claim. Applicant notes that on pages 16<sup>4</sup> – 17 of the Final Action, the Examiner suggests that “one of ordinary skill in the art will readily recognize that data is stored in a non-serialized format in a data store since data is returned to client from data store without serialization/deserialization”. Applicants submit that the Examiner’s reasoning is incorrect. Just because a distributed data manager and a client may communicate in-process configuration data without serialization/deserialization and may store data in its data store does not thereby render obvious the limitation that all data store operations performed by the distributed data manager in the in-process node store data in a non-serialized object format in a data store. The rejection of claim 6 is therefore further unsupported by the cited art, and removal thereof is respectfully requested.

With respect to claim 10, the Examiner first acknowledges that AAPA-Jin does not teach “the distributed data manager for the out-of-process node is configured to store the data in its serialized format”. On page 17 of the Final Action the Examiner asserts that “Crites teaches storing data in serialized format (Examiner has given a broadest reasonable interpretation of “data in serialized format” as “data stream” in view of specification (pages 3 lines 7 – 8). Crites teaches mass storage devices are stored a plurality of data streams, col. 2 lines 36 – 38. Therefore, storing data stream, disclosed by Crites, reads on storing data in serialized format.” The Examiner’s interpretation is incorrect. The cited portion of AAPA teaches that “serialization may include generating object data sequentially so that it may be transmitted as a data stream”, but this does not imply that storing a “continuous media data stream”, as taught in Crites, somehow represents data being stored in a serialized format. Furthermore, in response to Applicant’s argument that in Crites the “media stream” is sent to the client, whereas in claim 10, serialized data is sent from the client to a distributed data manager for storage, the Examiner cites portions of AAPA that also fail to teach the claimed limitation wherein the out-of-process client is configured to send serialized data to the distributed

data manager for the out-of-process node to store data. As pointed out in Applicant's previous response, even if the teachings of Crites were somehow to be combined with those of Jin and AAPA, the resulting system would merely allow media streams to be served from the nodes of a homogeneous distributed data system (i.e., one in which the nodes are either all "in-process nodes" or all "out-of-process nodes"), and would not render the combination of limitations of claim 10 obvious. Thus, the rejection of claim 10 is clearly unsupported by the cited art.

Applicant also asserts that numerous other ones of the dependent claims recite further distinctions over the cited art. However, since the rejection of the independent claims has been shown to be unsupported, a further discussion of the dependent claims is not necessary at this time.

## CONCLUSION

Applicants submit the application is in condition for allowance, and prompt notice to that effect is respectfully requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5681-05700/RCK.

Also enclosed herewith are the following items:

- ☒ Return Receipt Postcard
- ☐ Petition for Extension of Time
- ☐ Notice of Change of Address
- ☐ Other:

Respectfully submitted,



Robert C. Kowert  
Reg. No. 39,255  
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.  
P.O. Box 398  
Austin, TX 78767-0398  
Phone: (512) 853-8850

Date: May 22, 2006